

CLAIMS

What is claimed is:

- Sub  
A1
1. In a multithreaded computing environment, a method of processing computing tasks, comprising:
    - 5 defining a plurality of worker threads, each thread capable of processing a task;
    - defining a plurality of task queues, each task queue capable of queuing a plurality of tasks;
    - associating each task queue with a respective worker thread; and
    - 10 assigning a task to a task queue in an essentially random fashion.
  2. The method of Claim 1 wherein assigning a task comprises selecting an empty task queue.
  3. The method of Claim 2 wherein selecting comprises determining whether the selected task queue is in a busy state.
  - 15 4. The method of Claim 1 further comprising, from a worker thread, processing a task from the associated task queue.
  5. The method of Claim 1 further comprising, from a worker thread, processing a task from a task queue not associated with the thread.
  - 20 6. In a multithreaded computing environment, a method of processing computing threads, comprising:
    - defining a plurality of worker threads, each thread capable of processing a task;

-11-

ality of task queues, each task queue accessible by the worker threads; each task queue with a respective worker thread to an assigned task queue; and a thread not associated with the assigned task queue.

5 where assigning comprises selecting a pseudo-random number.

6 wherein assigning comprises selecting a pseudo-random number.

8 wherein selecting comprises determining a state.

computing environment, a system for providing a task queue to a worker thread, each thread capable of executing a task queue, each task queue capable of executing a task queue associated with a respective worker thread for a task to a task queue in an environment.

0 wherein the task scheduler selects a task queue to a worker thread.

1 wherein the task scheduler further selects a task queue to a worker thread when the worker thread is in a busy state.

assigning a task to an assigned task queue; and

5

7

8.

9.

10.

15

20

12

Sub A1

# THE LIFE OF

13. The system of Claim 10 further comprising a worker thread processing a task from the associated task queue.

14. The system of Claim 10 further comprising a worker thread processing a task from a task queue not associated with the thread.

5 15. In a multithreaded computing environment, a system for processing computing threads, comprising:

a plurality of worker threads, each thread capable of processing a task;

a plurality of task queues, each task queue capable of queuing a plurality

of tasks accessible by the worker threads and each task queue associated with a respective worker thread;

a task scheduler for assigning a task to an assigned task queue; and

wherein the assigned task is processed by a thread not associated with the assigned task queue.

16. The system of Claim 15 where the task scheduler selects the assigned task queue  
15 based on an essentially random number.

17. The system of Claim 15 wherein the task scheduler selects an empty task queue for assigning the task.

18. The system of Claim 17 wherein the task scheduler further determines whether the task queue is in a busy state.

20 19. An article of manufacturing, comprising:  
a computer-readable medium;

Sub  
A1

1992

5

Sub A1

10

20. The article of Claim 19 wherein the instructions for assigning a task comprise selecting an empty task queue.

15

22. The article of Claim 19 further comprising instructions for processing, in a worker thread, a task from the associated task queue.

23. The article of Claim 19 further comprising instructions for processing, in a worker thread, a task from a task queue not associated with the thread.

20 24. An article of manufacture, comprising:

a computer-readable medium;

a computer-implemented program for processing computing threads, in a multithreaded computing environment embodied in the medium, the program comprising instructions for:

